```
# Extract data

!unzip '/content/Flowers-Dataset.zip'

from tensorflow.keras.preprocessing.image import ImageDataGenerator
```

Image Augmentation

```
train_datagen = ImageDataGenerator(rescale=1./255,
                                   zoom_range=0.2,
                                   horizontal_flip=True)

xtrain = train_datagen.flow_from_directory('/content/flowers',
                                           target_size=(64,64),
                                           class_mode='categorical',
                                           batch_size=100)
```

Add Layers(Convolution, MaxPooling, Flatten, Dense-(Hidden Layers), Output)

```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Convolution2D, MaxPooling2D,
Flatten, Dense

model = Sequential() # create model sequence
model.add(Convolution2D(32,
(3,3),activation='relu',input_shape=(64,64,3)))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Flatten())
model.add(Dense(300,activation='relu'))
model.add(Dense(150,activation='relu'))
model.add(Dense(5,activation='softmax'))
```

Compile the Model

```
model.compile(optimizer='adam',loss='categorical_crossentropy',metrics
=['accuracy'])
```

Fit the Model

```
from tensorflow.keras.callbacks import EarlyStopping,
ReduceLROnPlateau

early_stop = EarlyStopping(monitor='val_accuracy',
                           patience=5)

lr = ReduceLROnPlateau(monitor='val_accuaracy',
                       factor=0.5,
                       min_lr=0.00001)
```

```
callback = [early_stop,lr]

model.fit_generator(xtrain,
                    steps_per_epoch=len(xtrain),
                    epochs=50,
                    callbacks=callback,
                    validation_data=xtest,
                    validation_steps=len(xtest))
```

Save the Model

```
model.save('flowers.h5')
```

Test the Model

```
from tensorflow.keras.preprocessing import image
import numpy as np

# Daisy

img =
image.load_img('/content/flowers/daisy/99306615_739eb94b9e_m.jpeg',tar
get_size=(64,64))
x = image.img_to_array(img)
x = np.expand_dims(x,axis=0)
pred = np.argmax(model.predict(x))
op = ['daisy','dandelion','rose','sunflower','tulip']
op[pred]

# Rose

img =
image.load_img('/content/flowers/rose/172311368_49412f881b.jpeg',targe
t_size=(64,64))
x = image.img_to_array(img)
x = np.expand_dims(x,axis=0)
pred = np.argmax(model.predict(x))
op = ['daisy','dandelion','rose','sunflower','tulip']
op[pred]

# Sunflower

img =
image.load_img('/content/flowers/sunflower/40410963_3ac280f23a_n.jpeg'
,target_size=(64,64))
x = image.img_to_array(img)
x = np.expand_dims(x,axis=0)
pred = np.argmax(model.predict(x))
```

```python
op = ['daisy','dandelion','rose','sunflower','tulip']
op[pred]

# Tulip

img = 
image.load_img('/content/flowers/tulip/110147301_ad921e2828.jpeg',targ
et_size=(64,64))
x = image.img_to_array(img)
x = np.expand_dims(x,axis=0)
pred = np.argmax(model.predict(x))
op = ['daisy','dandelion','rose','sunflower','tulip']
op[pred]
```